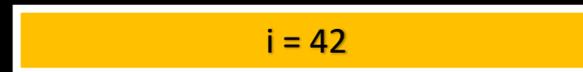


Programmieren II: Python Grundlagen

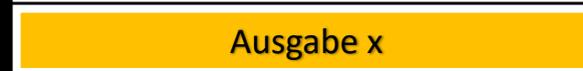
Aktueller Wissensstand

- Können mit **Struktogrammen** auf Papier programmieren
- Kennen die **wichtigsten Elemente** der Programmierung:

- **Variablen:**



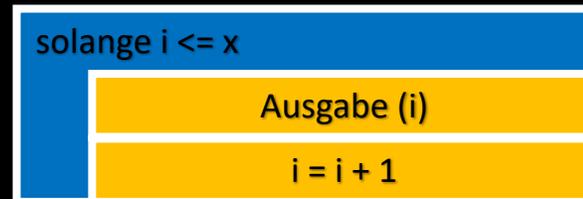
- **Eingaben & Ausgaben:**



- **Verzweigungen:**
(Entscheidungen)



- **Schleifen:**
(Codeblock wiederholen)



- Good News! Diese Elemente bilden die **Grundelemente (fast) aller Programmiersprachen!**
- Wollen nun **richtig Programmieren** (am Computer)

Welche Programmiersprache lernen?

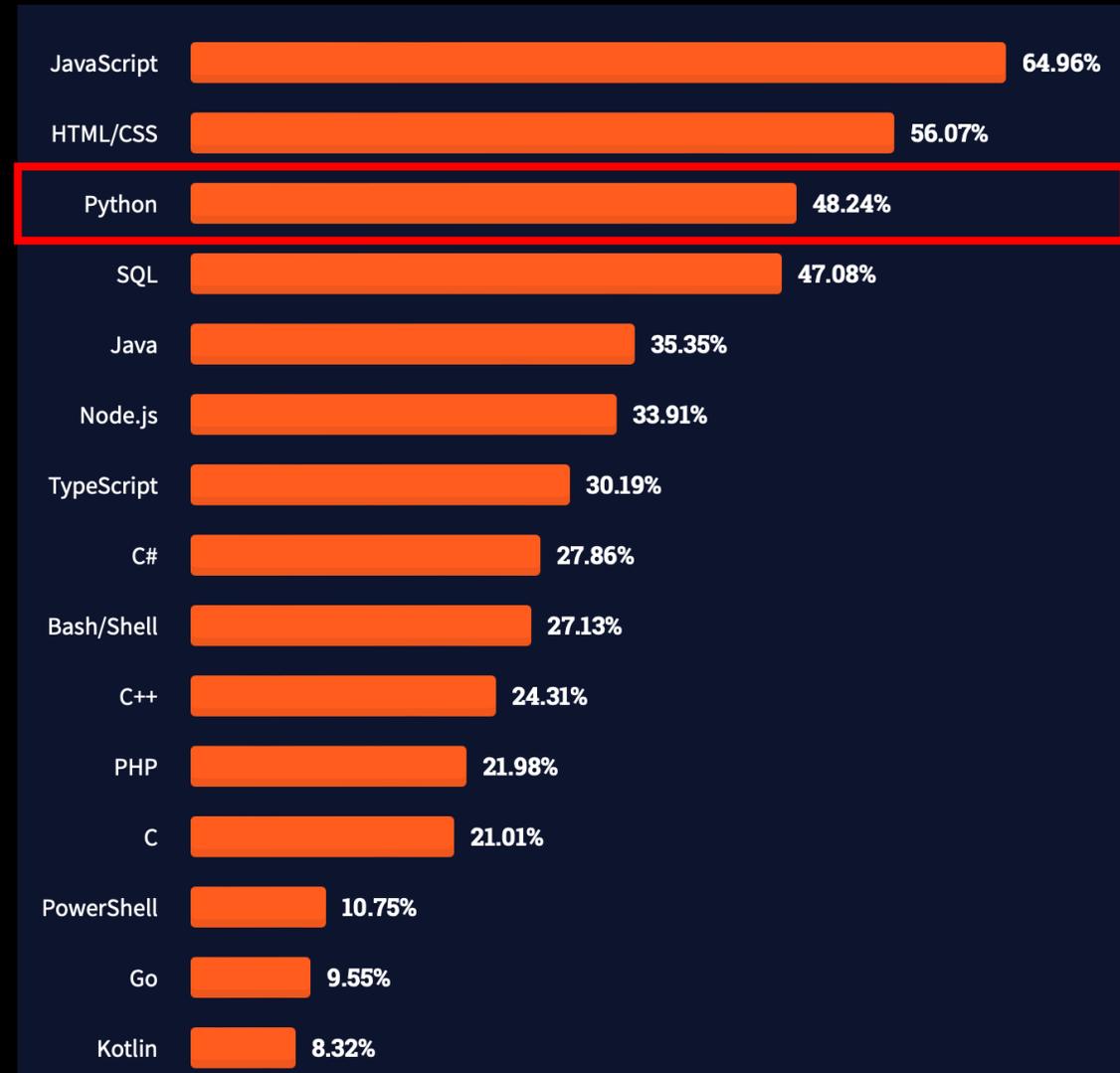
1. Anwendungsbereich?

- Wissenschaftliche Arbeiten
- Websites
- Games
- Desktopapplikationen (z.B. Word, Photoshop, ...)
- ...

2. Relevanz in 2022?

- Community
- Verfügbare Materialien (Tutorials, Videos, ...)

3. Schwierigkeitsgrad?



Warum Python?



- Einfache Programmiersprache
- Typischerweise in Top 3 nach Web-Sprachen (JavaScript, HTML/CSS)
- Wichtigste Programmiersprache 2022 in **Wissenschaft**
- Zitat aus Online-Referat von Physiker Prof. Dr. Andrew Lundgren @KSR



What is your number one advice for current students to be successful?

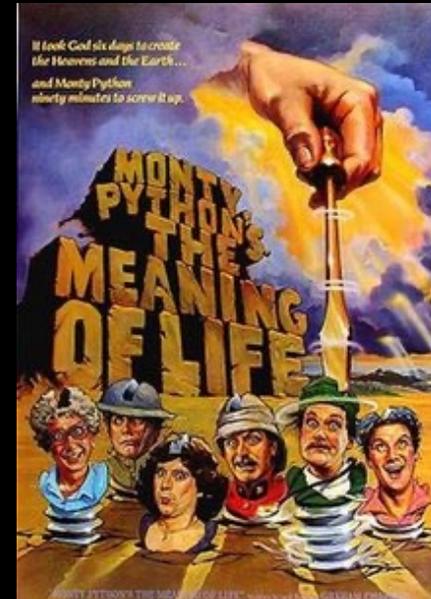
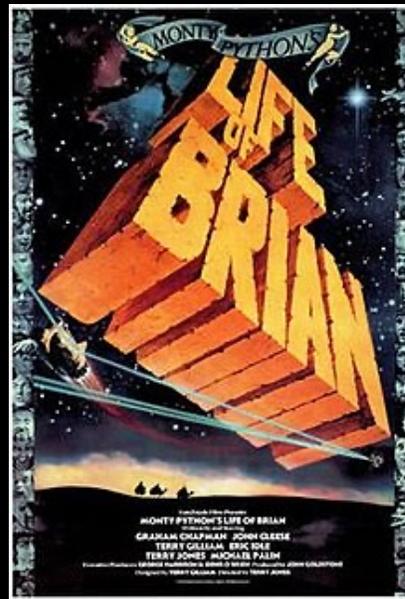
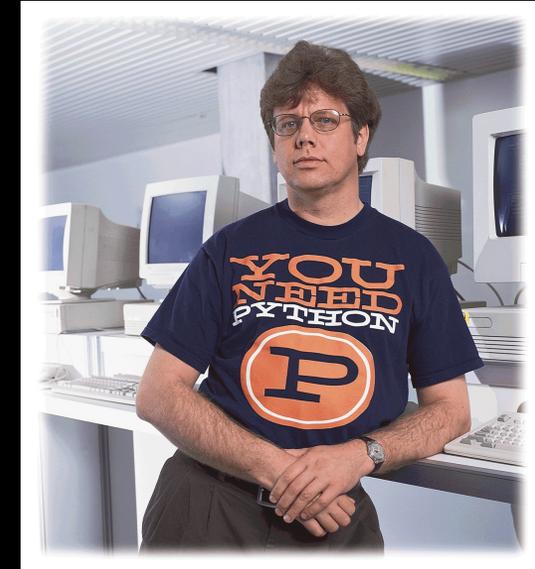
Learn Python!



Warum Python?



- Benannt nach **Monty Python!!!**
- Offizielle Python Website:
«When he began implementing Python, **Guido van Rossum** was also reading the published scripts from «Monty Python's Flying Circus», a BBC comedy series from the 1970s. Van Rossum thought he needed a name that was short, unique, and slightly mysterious, so he decided to call the language Python.»
- Film Empfehlungen:



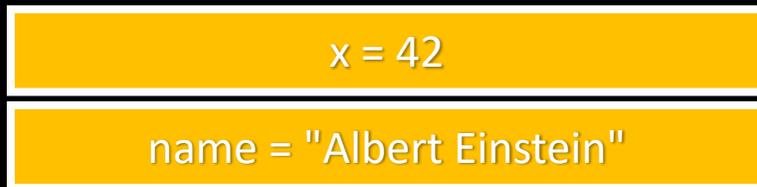
Von Struktogrammen zu Python

- Alle Elemente von Struktogrammen gibt es in ganz ähnlicher Form auch in Python
- Wollen nun 'Übersetzen': Struktogramme -> Python

Variablen

- In einer **Variablen** kann ein Wert *gespeichert* werden. Zum Beispiel kann man der Variablen **x** den Wert **42** oder der Variablen **name** den Wert **"Albert Einstein"** zuweisen.

- Struktogramm:



- Python:

```
x = 42
name = "Albert Einstein"
```

- Variablen haben unterschiedliche **Datentypen**:

- **x** ist ein **Integer** (kurz **int**), also eine ganze Zahl
- **name** ist ein **String**, also Text. Dafür werden Anführungs- & Schlusszeichen verwendet.

Eingabe & Ausgabe

- Eingabe: Benutzer:in kann Wert eingeben
- Ausgabe: Wert (z.B. Resultat) kann ausgegeben/angezeigt werden
- Struktogramm:



Python:

```
x = int(input("Gib eine Zahl ein"))  
s = str(input("Gib einen String ein"))  
print(x)
```

- `input()`:
 - Eingabe/Input von Benutzer:in verlangen
 - Kann in runden Klammern **Mitteilung** schreiben
 - `int` steht für Integer, also ganze Zahl (...,-2,-1,0,1,2,...)
 - Mit `int()` kann sicherstellen, dass Eingabe als Zahl (und nicht Text) aufgefasst wird
 - Mit `str()` kann sicherstellen, dass Eingabe als Text/String (nicht Zahl) aufgefasst wird

Schleifen

- Mit einer **Schleife** kann man einen Codeblock wiederholt ausführen lassen und zwar solange, wie eine vorgegebene *Bedingung erfüllt* ist.

- Struktogramm:



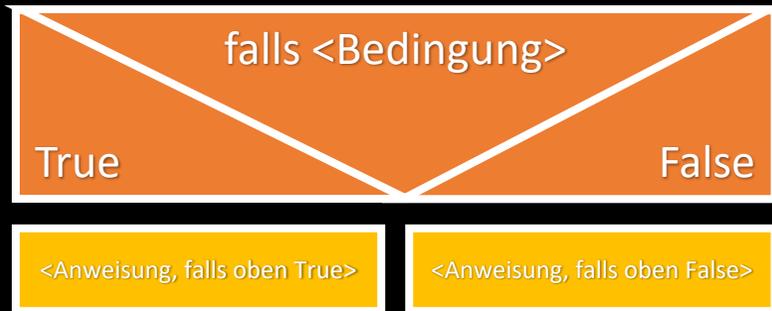
- Python:

```
while <Bedingung>:  
    # Anweisung 1  
    # Anweisung 2  
    # ...
```

- Python: Codeblock, der wiederholt werden soll, wird mit TAB-Taste **eingerrückt**
- **Vergleichsoperatoren** in Bedingungen: `==`, `<`, `>`, `<=`, `>=`, `!=` (ungleich)
- Zeilen die mit `#` beginnen sind **Kommentare** (kein Code)

Verzweigung

- Je nach Situation, nimmt Code anderen Verlauf
- Struktogramm: Python:



```
if <Bedingung>:  
    # Codeblock der ausgeführt wird,  
    # falls (if) Bedingung erfüllt (True) ist  
else:  
    # Codeblock der ausgeführt wird,  
    # der ansonsten ausgeführt wird,  
    # falls also Bedingung NICHT erfüllt ist (False)
```

- Wichtig: korrekte **Einrückungen** mit TAB wie bei Schleifen

Aufgaben A

- Idee:
 - dir bekannte Struktogramme auf Python übersetzen (auf Papier)
 - ... und dann am Computer ausprobieren